REMARKS

Claims 1-16 were rejected in the parent application. For the current application, the applicant has amended claims 1-3, 6, and 11-16 and added new claims 17-28.

The applicant has amended claim 1, which now recites: "A computer-based method of processing a computer graphics illustration that includes one or more pieces of artwork, the method comprising: mapping outlines of at least one of the pieces of artwork onto a grid of cells; determining the total number of outlines of pieces of artwork that map to a cell of the grid; and identifying the cell as a complex region based on the total number of outlines that map to the cell."

The applicant respectfully submits that Schiller fails to disclose or suggest at least one element of claim 1. For example, claim 1 recites: "determining the total number of outlines of pieces of artwork that map to a cell of the grid; and identifying the cell as a complex region based on the total number of outlines that map to the cell."

Lines 43-47 of column 6 of Schiller describe a method that is significantly different from the one recited by claim 1. This passage describes assigning a complexity value to a tile based on "the number of path intersections occurring within its boundary plus the number of local minima and maxima that occur within the boundary." *See* Col. 6, lines 45-48. As specified by the passage, it is the number of path intersections that occur within the boundary of a tile that determines the complexity value of the tile. In sharp contrast, the determination recited by claim 1 is not based on the number of path intersections that occur within a cell but rather the total number of outlines of pieces of artwork that map to the cell.

An example using figures in the instant application will illustrate the difference. Figure 18 of the application shows several outlines being mapped onto a grid. One of the outlines, the one of artwork 138 is completely inside one of the grids cells. Figure 19 shows the complexity value of each of the grid cells. Cell 146 has a complexity value of "4" since four outlines map to the cell. These are the outlines of pieces of artwork 136, 138, 140 and 142. It is important to note that none of the outlines in cell 146 actually intersect; the complexity value is due merely to their presence in the cell. In contrast, the cited passage of Schiller describes a method that would assign a complexity value of "0" to the 146 cell because the cell contains no path intersections.

Arguments along these lines were made in the applicant's first response filed December 11, 2001. In response, the Examiner points to column 1, lines 46-48 of Schiller, which states that "if more than two objects (paths) are present, the graphical processing system must be able to determine which parts of which paths overlap." This passage describes a requirement of a graphical processing system and has nothing to do with determining complexity value of a cell.

The Examiner also points to lines 6-13 of column 7, which describe a memory reallocation process. The applicant submits that this passage also has nothing to do with determining complexity value of a cell. Thus, the applicant respectfully submits these passages of Schiller and the one cited above do not disclose or suggest the quoted element of claim 1.

Lines 43-47 of column 6 of Schiller also fail to disclose or suggest the quoted element of claim 1. This passage describes assigning complexity value based on the number of path intersections within a boundary of a tile and is significantly different than the quoted element of claim 1. The Examiner reads this passage as disclosing assigning a complexity value to a tile based on the number of path intersections with the tile boundary. The applicant notes that this appears to misread the passage in question, but, even if the Examiner were correct the passage would not disclose or suggest the quoted element of the claim which is directed to determining and assigning complexity based on the total number of outlines that map to a cell, and not on a number of path intersections, whether within a tile or with a tile boundary. Another example will illustrate. Take, for example, a grid with a circle that is inside of, but which does not intersect with, the grid's boundary. The Schiller method, as construed by the Examiner, would assign a complexity value of "0" because there is no intersection between the path of the circle and the boundary of the tile. In contrast, the method of claim 1 would assign a complexity value of "1" because the circle maps to the cell. Thus, the applicant respectfully submits that this cited passage of Schiller fails to disclose or suggest the quoted element of claim 1.

The Examiner suggests that another passage of Schiller, i.e., lines 51-54 of column 6, discloses the quoted element of claim 1. Lines 51-54 of column 6 read: "As each path (or pair of paths as described above) is processed at step 715, the complexity value of all those tiles intersected by the path are updated (step 720)." The applicant respectfully points out that this passage describes when the complexity value of a given tile is calculated and not the manner in which the complexity value is calculated. Thus, the applicant submits that this passage, like the ones cited above, also does not disclose or suggest the quoted element of claim 1.

For at least the above reasons, Schiller does not disclose or suggest the limitations of claim 1. The applicant respectfully submits that claim 1 and claims 2-14, which depend from claim 1, are in condition for allowance.

The applicant has amended claim 15, which now recites: "A computer program product, tangibly stored on machine-readable medium, for processing a computer graphics illustration having pieces of artwork, the product comprising instructions operable to cause a processor to: map outlines of at least one of the pieces of artwork onto a grid of cells; determine the total number of outlines of pieces of artwork that map to a cell of the grid; and identify the cell as a complex region based on the total number of outlines that map to the cell." Claim 15 includes an element that is similar to the quoted claim 1 element and the foregoing remarks made with respect to claim 1 apply with equal for to claim 15. For at least this reason, the applicant respectfully submits that claim 15 and claims 16-20, which depend from claim 15, are in condition for allowance.

Attached is a marked-up version of the changes being made by the current amendment. The applicant asks that all claims be examined. Enclosed is $110 check for a one-month extension of time. Please apply any other appropriate charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: January 3, 2003

Tim H. Pham
Reg. No. 48,589

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

Version with markings to show changes made

In the claims:

Claims 1-3, 6, and 11-16 have been amended as follows:

1. (Amended) A computer-based method of processing a computer graphics illustration [having] that includes one or more pieces of artwork, the method comprising:

mapping outlines of at least [some] one of the pieces of artwork onto a grid of cells;

determining [a] the total number of outlines of pieces of artwork that map to a cell of the grid; and

identifying the cell as a complex region based on the [determined] total number of outlines that map to the cell.

2. (Amended) The method of claim 1, further comprising identifying [artwork] pieces of artwork to include in an illustration flattening process based on the identification of the complex region.

3. (Amended) The method of claim 2 wherein [an] the illustration flattening process comprises a process for producing a planar map [from an] of the illustration.

6. (Amended) The method of claim 1 wherein identifying comprises comparing the [determined] total number of [artwork pieces that enter a] outlines of pieces of artwork that map to the cell with a threshold.

11. (Amended) The method of claim 1 further comprising classifying at least one of the pieces of artwork based on the intersection of the piece of artwork with the complex region[s].

12. (Amended) The method of claim 11 wherein classifying comprises identifying the piece of artwork as being completely inside [a] the complex region.

13. (Amended) The method of claim 11 wherein classifying comprises identifying the piece of artwork as being completely outside [a] the complex region.

14. (Amended) The method of claim 11 wherein classifying comprises identifying the piece of artwork as being partially inside [a] the complex region.

15. (Amended) A computer program product, [disposed on a computer] tangibly stored on machine-readable medium, for processing a computer graphics illustration having pieces of artwork, the [computer program] product comprising instructions [for causing] operable to cause a processor to:

map outlines of at least [some] one of the pieces of artwork onto a grid of cells;

determine [a] the total number of outlines of pieces of artwork that map to a cell of the grid; and

identify the cell as a complex region based on the [determined] total number of outlines that map to the cell[; and]

[based on the identifying, excluding pieces of artwork from an illustration flattening process].

16. (Amended) The computer program of claim 17, [15 wherein excluding pieces of artwork comprises excluding] further comprising instructions to:

exclude pieces of artwork [classified as] that map entirely inside the complex region.